

# Structural index reduction algorithms for differential algebraic equations via fixed-point iteration

Juan Tang<sup>\*</sup>    Wenyuan Wu<sup>†</sup>    Xiaolin Qin<sup>‡</sup>    Yong Feng<sup>§</sup>

## Abstract

Motivated by Pryce's structural index reduction method for differential algebraic equations (DAEs), we show the complexity of the fixed-point iteration algorithm and propose a fixed-point iteration method with parameters. It leads to a block fixed-point iteration method which can be applied to large-scale DAEs with block upper triangular structure. Moreover, its complexity analysis is also given in this paper.

**Keywords:** differential algebraic equations, structural analysis, index reduction, linear programming, fixed-point iteration, block triangular forms.

**MSC(2010):** 34A09, 65L80, 65F50, 90C05, 90C27, 90C06.

## 1 Introduction

Differential algebraic equations (DAEs) systems arise naturally in modeling many dynamical systems, such as electric circuits, mechanical systems, and

---

<sup>\*</sup>Chengdu Institute of Computer Applications, Chinese Academy of Sciences (CAS); *Emails:* tj-123123@163.com, tangjuan0822@gmail.com

<sup>†</sup>Automated Reasoning and Cognition Key Lab of Chongqing, CIGIT, CAS; *Email:* wuwenyuan@cigit.ac.cn.

<sup>‡</sup>Chengdu Institute of Computer Applications, Chinese Academy of Sciences; *Email:* qinxl@casit.ac.cn

<sup>§</sup>Corresponding author. Automated Reasoning and Cognition Key Lab of Chongqing, CIGIT, CAS; *Email:* yongfeng@cigit.ac.cn.

spacecraft dynamics. Based on unified multi-domain modeling techniques e.g. Modelica [9], computers can automatically produce thousands of DAEs. The generated DAEs have many interesting characteristics, such as large scale, high index, block structures, which are the major motivations of our work in this paper. It is well known that a direct numerical simulation without index reduction may not be possible or may provide a bad result [12, 5]. Here the index of a DAE system is a key notion in the theory for measuring the distance from the given system with a singular Jacobian to the corresponding ordinary differential equations with a nonsingular Jacobian. Various index concepts exist in the theory of DAEs; and the one related to the structural analysis approach is the “*structural index*”, which is defined in (5). For other indices, we refer the interested readers to [1, 7, 18]. High-index DAE systems usually need differentiations to reveal all the system’s constraints, which are crucial to determine consistent initial conditions. This procedure is the called “*index reduction*” of DAEs. For applications of high index DAEs, see [16]. Identifying all hidden constraints on formal power series solutions in the neighborhood of a given point is a key step to construct nonsingular Jacobian of a DAE system for numerical integration. Thus, for DAE systems, index reduction is fundamental and unavoidable.

In the previous work on DAE index reduction for general DAE systems, Campbell and Gear gave a derivative-array method to reduce DAEs in [2], which may not be applicable to large-scale nonlinear systems. Pantelides in [11] introduced a graph-oriented method which gives a systematic way to reduce high-index of DAEs with order one to lower index, by selectively adding differentiated forms of the equations already present in the system. In [14], Pryce developed structural analysis method which is proved to computes the same structural index with Pantelides’ method and is straightforward method for analyzing the structure of DAEs of any order. This approach is based on solving an assignment problem, which can be formulated as an integer linear programming problem. The idea was generalized to a class of partial differential algebraic equations by Wu et al [20]. Recently, Pryce et al. in [15, 10] generalized the structural analysis method to the DAE systems with *coarse* or *fine* block triangular forms (BTF), and showed that the difference between global offsets of signature matrix ( $\Sigma$ ) and local offsets of each sub-block in  $\Sigma$  with *fine* BTF is constant. We focus on structural index reduction method for block triangularized systems to directly calculate the smallest offsets of the system in sequential block-wise manner, and give the complexity analysis of structural index reduction algorithms for DAEs

systems without or with BTF.

The rest of this paper is organized as follows. Section 2 briefly reviews Pryce's structural analysis method, firstly. Then we novelly and finely prove the existence and uniqueness of smallest optimal solution of Problem 4 and show the termination of fixed-point iteration algorithm. In addition, we also give the time complexity of the algorithm which is  $O(n^3 + \|\mathbf{c}^*\|_1 \cdot n^2)$  due to Theorem 2.6 and is not given in [14], where  $n$  is the size of the system. Section 3 first introduces the block triangular forms (BTF) for large scale DAE systems. Based on our fixed-point iteration method with parameter, a block fixed-point iteration algorithm is proposed to find the unique smallest dual-optimal pair of the systems with BTF, and its time complexity is  $O(\sum_{i=1}^{\ell} n_i^3 + \|\mathbf{c}_i^*\|_1 \cdot n_i^2)$  by Theorem 3.5, where  $\sum_{i=1}^{\ell} n_i = n$  and  $\ell$  is the number of the blocks on the diagonal. It is usually much better than the cost  $O(n^3 + \|\mathbf{c}^*\|_1 \cdot n^2)$  without taking the advantages of the structure, when  $\ell$  is large. Conclusions are made in the last section.

## 2 Theoretical foundation for fixed-point iteration method

First we give a brief review about the main steps of Pryce's structural analysis method [14]. We consider a DAE system  $\mathbf{f} = (f_1, f_2, \dots, f_n) = \mathbf{0}$  in  $n$  dependent variables  $x_j = x_j(t)$  with  $t$  a scalar independent variable, of the form

$$f_i = f_i(t, \text{the } x_j \text{ and derivatives of them}), \quad i = 1, 2, \dots, n. \quad (1)$$

Step 1. Form the  $n \times n$  signature matrix  $\Sigma = (\sigma_{ij})$  of the DAE, where

$$\sigma_{ij} = \begin{cases} \text{highest differential order of } x_j \text{ in equation } f_i, & \text{if } x_j \text{ appears in } f_i, \\ -\infty, & \text{otherwise.} \end{cases}$$

Step 2. Solve an assignment problem (AP) to find a highest value transversal (HVT)  $T$ , which is a subset of sparsity pattern  $S$  with  $n$  finite entries and describes just one element in each row and each column, such that  $\sum \sigma_{ij}$  is maximized and finite. The sparsity pattern  $S$  of  $\Sigma$  is defined as:

$$S = \text{sparse}(\Sigma) = \{(i, j) : \sigma_{ij} > -\infty\}. \quad (2)$$

This can be formulated as a Linear Programming Problem (LPP), the Primal is:

$$\begin{aligned}
\max_{\xi} \quad & z = \sum_{(i,j) \in S} \sigma_{ij} \xi_{ij}, \\
\text{s.t.} \quad & \sum_{j: (i,j) \in S} \xi_{ij} = 1 \text{ for each } i, \\
& \sum_{i: (i,j) \in S} \xi_{ij} = 1 \text{ for each } j, \\
& \xi_{ij} \geq 0 \text{ for } (i,j) \in S.
\end{aligned} \tag{3}$$

The problem is equivalent to finding a maximum-weight perfect matching in a bipartite graph whose incidence matrix is the signature matrix, and can be solved by Kuhn-Munkres algorithm[19] whose time complexity is  $O(n^3)$ .

Step 3. Determine the offsets of the problem, which are the vectors  $\mathbf{c} = (c_i)_{1 \leq i \leq n}$ ,  $\mathbf{d} = (d_j)_{1 \leq j \leq n}$ , the smallest such that  $d_j - c_i \geq \sigma_{ij}$ , for all  $1 \leq i \leq n, 1 \leq j \leq n$ , and  $d_j - c_i = \sigma_{ij}$  when  $(i,j) \in T$ .

This problem can be formulated as the dual of (3) in the variables  $\mathbf{c} = (c_1, c_2, \dots, c_n)$  and  $\mathbf{d} = (d_1, d_2, \dots, d_n)$ . The Dual is defined as follows:

$$\begin{aligned}
\min_{\mathbf{c}, \mathbf{d}} \quad & z = \sum_j d_j - \sum_i c_i, \\
\text{s.t.} \quad & d_j - c_i \geq \sigma_{ij} \text{ for all } (i,j), \\
& c_i \geq 0 \text{ for all } i.
\end{aligned} \tag{4}$$

Step 4. Form the system Jacobian matrix  $\mathbf{J}$ , given by

$$\mathbf{J}_{ij} = \begin{cases} \frac{\partial f_i}{\partial ((d_j - c_i) \text{th derivative of } x_j)}, & \text{if this derivative is present in } f_i, \\ 0, & \text{otherwise.} \end{cases}$$

Step 5. Choose a consistent point. If  $\mathbf{J}$  is non-singular at that point, then the solution can be computed with Taylor series or numerical homotopy continuation techniques in a neighborhood of that point. And using the smallest offsets  $\mathbf{c}, \mathbf{d}$  of Problem 4, the structural index is then defined as:

$$\nu = \max_i c_i + \begin{cases} 0, & \text{for all } d_j > 0, \\ 1, & \text{for some } d_j = 0. \end{cases} \tag{5}$$

In order to determine the smallest offsets of DAEs using fixed-point iteration algorithm [14], we introduce some necessary definitions, firstly. Define a natural semi-ordering of vectors in  $\mathbb{R}^n$ , for  $\forall \mathbf{a}, \mathbf{b}$ ,  $\mathbf{a} \prec \mathbf{b}$  if  $a_i \leq b_i$  for each

$i$ , smallest of offsets is in the sense of ordering  $\prec$ . Given  $\Sigma$  of DAEs systems and a corresponding transversal  $T$ , for  $\forall \mathbf{c} = (c_i) (\in \mathbb{R}^n)$ , we define a mapping

$$\mathcal{D}(\mathbf{c}) = (d_j), \text{ where } d_j = \max_i (\sigma_{ij} + c_i),$$

and for  $\forall \mathbf{d} = (d_j) (\in \mathbb{R}^n)$ , we define a mapping

$$\mathcal{C}_T(\mathbf{d}) = (c_i^*), \text{ where } c_i^* = d_j - \sigma_{i,j}, (i,j) \in T.$$

Furthermore, we define the composition mapping  $\phi_T(\mathbf{c}) = \mathcal{C}_T(\mathcal{D}(\mathbf{c}))$  from  $\mathbb{R}^n$  to  $\mathbb{R}^n$ . Then we obtain fixed-point iteration algorithm below.

---

**Algorithm 1** Fixed-point iteration algorithm

---

**Input:**

$\Sigma$  is the signature matrix of DAEs

**Output:**

$\mathbf{c}$  and  $\mathbf{d}$

- 1: Set  $\mathbf{c}' \leftarrow \mathbf{0}$
  - 2: Set  $\mathbf{d} \leftarrow \mathcal{D}(\mathbf{c}')$
  - 3: Set  $T$  is the HVT of  $\Sigma$  computed by Kuhn-Munkres algorithm
  - 4: Set  $\mathbf{c} \leftarrow \mathcal{C}_T(\mathbf{d})$
  - 5: **while**  $\mathbf{c} \neq \mathbf{c}'$  **do**
  - 6:   Set  $\mathbf{c}' \leftarrow \mathbf{c}$
  - 7:   Set  $\mathbf{d} \leftarrow \mathcal{D}(\mathbf{c})$
  - 8:   Set  $\mathbf{c} \leftarrow \mathcal{C}_T(\mathbf{d})$
  - 9: **end while**
  - 10: **return**  $\mathbf{c}, \mathbf{d}$
- 

In order to give a novel and refined proof of the existence and uniqueness of smallest offsets of Problem 4 and the termination of Algorithm 1, we introduce some definitions and lemmas as follows.

**Lemma 2.1** ([14]).

- (i) *If the Primal has a feasible solution, then it has a basic feasible solution (BFS). At such a solution the  $\xi_{ij}$  are 1 on some transversal  $T$  and 0 elsewhere. The corresponding objective function value is  $\sum_{(i,j) \in T} \sigma_{ij}$ , denoted by  $\|T\|$ . The optimum is achieved at a BFS.*
- (ii) *The following results are equivalent.*

- (a) The AP is regular.
- (b) The Primal has a feasible solution.
- (c) Primal and Dual have a common, finite optimal value given by

$$z = \sum d_j - \sum c_i = ||T||. \quad (6)$$

(iii)(Principle of complementary slackness) Given a Primal BFS, i.e., a transversal  $T$ , and a Dual feasible solution  $\mathbf{c}, \mathbf{d}$ , the following are equivalent:

- (a)  $T$  is an HVT, and  $\mathbf{c}$  and  $\mathbf{d}$  are optimal for the Dual.
- (b)  $d_j - c_i = \sigma_{ij}$ , for each  $(i, j) \in T$ .

**Lemma 2.2** ([14]). Assume that  $T$  is HVT, then  $\mathbf{c}$  is optimal of Problem 4 if and only if  $\mathbf{c}$  is the non-negative fix-points of  $\phi_T$ , that is,  $\phi_T(\mathbf{c}) = \mathbf{c}$ .

**Definition 2.1** For a given  $\Sigma$  and a corresponding  $T$ , the vector set  $VC$  is defined as:

$$VC(T) = \{\mathbf{c} \in \mathbb{R}^n | \phi_T(\mathbf{c}) = \mathbf{c}, \mathbf{c} \succ \mathbf{0}\}. \quad (7)$$

If  $T$  is HVT, the part optimal solution set of Problem 4 is  $VC(T)$  by Lemma 2.2. Furthermore, we have:

**Lemma 2.3** For a given  $\Sigma$  matrix, the optimal-dual set  $VC$  is independent of the choice of HVT, that is,  $VC(T) = VC(T')$  for any two HVT  $T$  and  $T'$ .

PROOF. For any  $\mathbf{c} \in VC(T)$ , define  $\mathbf{d} = \mathcal{D}(\mathbf{c})$ . Then  $\mathbf{c}$  and  $\mathbf{d}$  are optimal-dual by Lemma 2.2. Note that  $T'$  is HVT. By Lemma 2.1(iii), obtain

$$d_j - c_i = \sigma_{ij}, \quad (8)$$

for each  $(i, j) \in T'$ . According to (8), we have

$$\mathbf{c} = \mathcal{C}_{T'}(\mathbf{d}) = \mathcal{C}_{T'}(\mathcal{D}(\mathbf{c})) = \phi_{T'}(\mathbf{c}). \quad (9)$$

Therefore,  $\mathbf{c}$  is also the non-negative fix-points of  $\phi_{T'}$ , that is,  $\mathbf{c} \in VC(T')$ . Conversely, we can easily prove  $VC(T') \subseteq VC(T)$  with the similar principle above.  $\square$

Now, using the above results we can prove the existence and uniqueness of the smallest optimal solution for the Dual problem.

**Lemma 2.4** *Assume that the  $\Sigma$  matrix of given DAE systems in Problem 4 contains a transversal  $T$  at least, then there exists a unique smallest dual-optimal pair  $\mathbf{c}^*$  and  $\mathbf{d}^*$ .*

PROOF. The  $\Sigma$  matrix contains a transversal  $T$  at least, then there must exist a HVT  $T$  from the finiteness of transversal. From the Lemma 2.3, assume  $T$  is any HVT. According to the primal-dual principle, dual-optimal pair  $\mathbf{c}$  and  $\mathbf{d}$  must exist, that is,  $VC = VC(T)$  is a non-empty set by Lemma 2.2. Moreover, It is easy to know that for any non-negative vector  $\Theta = (\theta, \theta, \dots, \theta)$ ,  $\mathbf{c} + \Theta$  and  $\mathbf{d} + \Theta$  is also dual-optimal. Then  $VC$  is a infinite set. Define

$$VC_1 = \{\|\mathbf{c}\|_1 : \mathbf{c} \in VC\} \quad \text{and} \quad \alpha = \inf\{VC_1\}(\geq 0). \quad (10)$$

(The existence of ‘smallest’ dual-optimal in ordering ‘ $\leq$ ’) In fact, the coefficients of all the constraint equations in Problem 4 are 1 or  $-1$ , and each  $\sigma_{ij} \in \Sigma$  is integer. Thus, all the vertices of the feasible region in Linear programming are integer. Then the ‘smallest’ dual-optimal also are integer. Therefore, there exists  $\mathbf{c}^* \in VC$  such that  $\alpha = \|\mathbf{c}^*\|_1$ , that is,  $\mathbf{c}^*$  and  $\mathbf{d}^* = \mathcal{D}(\mathbf{c}^*)$  is the ‘smallest’ dual-optimal in the sense of ordering ‘ $\leq$ ’.

(The uniqueness of ‘smallest’ dual-optimal in ordering ‘ $\leq$ ’) Assume that there are two different ‘smallest’ dual-optimal pair  $\mathbf{c}^*, \mathbf{d}^*$  and  $\mathbf{c}^o, \mathbf{d}^o$  such that

$$\alpha = \|\mathbf{c}^*\|_1 = \|\mathbf{c}^o\|_1. \quad (11)$$

There must exist  $i_0 \in \{1, 2, \dots, n\}$  such that  $c_{i_0}^* \neq c_{i_0}^o$ . According to the following rules, construct vector pair  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$ . For given HVT  $T$  and each  $i$ , if  $c_i^* \geq c_i^o$ , define  $c_i^{*o} = c_i^o$  and  $d_j^{*o} = d_j^o$  such that  $(i, j) \in T$ ; otherwise, define  $c_i^{*o} = c_i^*$  and  $d_j^{*o} = d_j^*$ .

Firstly, it is verified that  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$  are the Dual feasible solution. By the definition of  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$ , for each  $j \in \{1, 2, \dots, n\}$ , obtain

$$d_j^{*o} = d_j^* \geq c_i^* + \sigma_{ij} \geq c_i^{*o} + \sigma_{ij}, i = 1, 2, \dots, n; \quad (12)$$

and

$$d_j^{*o} = d_j^o \geq c_i^o + \sigma_{ij} \geq c_i^{*o} + \sigma_{ij}, i = 1, 2, \dots, n. \quad (13)$$

Then together with (12) and (13), for each  $j \in \{1, 2, \dots, n\}$ , get

$$d_j^{*o} \geq c_i^{*o} + \sigma_{ij}, i = 1, 2, \dots, n. \quad (14)$$

That is  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$  are the Dual feasible solution.

Furthermore, note that T is HVT,  $\mathbf{c}^*$ ,  $\mathbf{d}^*$  and  $\mathbf{c}^o$ ,  $\mathbf{d}^o$  are dual-optimal pair. By Lemma 2.1(iii), obtain

$$d_j^* - c_i^* = \sigma_{ij}, \quad d_j^o - c_i^o = \sigma_{ij}, \quad (15)$$

for each  $(i, j) \in T$ . By (15), we have

$$d_j^{*o} - c_i^{*o} = \sigma_{ij}, \quad (16)$$

for each  $(i, j) \in T$ . Combining (14) and (16), it is indicated that  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$  are also the dual-optimal by Lemma 2.1(iii), that is,  $\mathbf{c}^{*o} \in VC$ . But  $\|\mathbf{c}^{*o}\|_1 < \|\mathbf{c}^*\|_1 = \alpha$  or  $\|\mathbf{c}^{*o}\|_1 < \|\mathbf{c}^o\|_1 = \alpha$ , which is in conflict with (10). Therefore, the smallest dual-optimal is unique in the sense of ordering ' $\leq$ '.

(Smallest in ordering  $\prec$ ) Set  $\mathbf{c}^*$  and  $\mathbf{d}^*$  are the 'smallest' dual-optimal,  $\mathbf{c}$  and  $\mathbf{d}$  are any dual-optimal, then obtain  $\alpha = \|\mathbf{c}^*\|_1 \leq \|\mathbf{c}\|_1$ . Assume that there exists  $i_0 \in \{1, 2, \dots, n\}$  such that  $c_{i_0}^* > c_{i_0}$ . We can construct the new dual-optimal  $\mathbf{c}^{o*} (\in VC)$  and  $\mathbf{d}^{o*}$  such that  $\|\mathbf{c}^{o*}\|_1 < \|\mathbf{c}^*\|_1 = \alpha$  by the method described above, which is also in conflict with (10). So obtain  $\mathbf{c}^* \prec \mathbf{c}$ , and  $\mathbf{d}^* = \mathcal{D}(\mathbf{c}^*) \prec \mathcal{D}(\mathbf{c}) = \mathbf{d}$ . Therefore,  $\mathbf{c}^*$  and  $\mathbf{d}^*$  is the unique smallest dual-optimal pair.  $\square$

According to the above lemmas, we can prove the termination of fixed-point iteration algorithm and analyze its complexity.

**Lemma 2.5** *The fixed-point iteration algorithm can find the unique smallest dual-optimal pair  $\mathbf{c}^*$  and  $\mathbf{d}^*$  of Problem 4 by at most  $\|\mathbf{c}^*\|_1 + 1$  iterations.*

PROOF. Set  $\mathbf{c}^{(1)} = \phi(\mathbf{0}) (\succ \mathbf{0})$ ,  $\mathbf{c}^{(k)} = \phi(\mathbf{c}^{(k-1)}) = \phi^k(\mathbf{0})$  for  $k \in \mathbb{N}^+$ . It is verified that  $\phi (= \phi_T)$  is monotone operator from  $\mathbb{R}^n$  to  $\mathbb{R}^n$ , that is, if  $\mathbf{c} \prec \mathbf{c}'$ ,  $\phi(\mathbf{c}) \prec \phi(\mathbf{c}')$ . So  $\{\mathbf{c}^{(k)}\}$  is a increasing sequence in " $\prec$ " sense, and  $\{\|\mathbf{c}^{(k)}\|_1\}$  is also a increasing sequence. Note that T is HVT, then exist the unique smallest dual-optimal pair  $\mathbf{c}^* (\succ \mathbf{0})$  and  $\mathbf{d}^*$  by Lemma 2.4. According to the monotonicity of  $\phi$ , obtain

$$\mathbf{c}^{(k)} \prec \mathbf{c}^*, \text{ for } k \in \mathbb{N}^+, \quad (17)$$

and then  $\|\mathbf{c}^{(k)}\|_1 \leq \|\mathbf{c}^*\|_1$ . It is indicated that  $\{\|\mathbf{c}^{(k)}\|_1\}$  is bounded. Based on bounded monotonic principle, exists  $\beta$  such that

$$\|\mathbf{c}^{(k)}\|_1 \rightarrow \beta (\leq \|\mathbf{c}^*\|_1), k \rightarrow \infty. \quad (18)$$



Assume  $\{\|\mathbf{c}^{(k)}\|_1\}$  is strictly increasing sequence. For all  $c_i^{(k)}$  are integer, we have  $\|\mathbf{c}^{(k)}\|_1 \geq k$ . So  $\beta = +\infty$ , which is in conflict with (18). It is shown that exists  $m \in \mathbb{N}^+$  such that  $\|\mathbf{c}^{(m)}\|_1 = \|\mathbf{c}^{(m+1)}\|_1$ . Moreover, obtain  $\mathbf{c}^{(m)} = \mathbf{c}^{(m+1)}$  and  $\mathbf{c}^{(l)} = \mathbf{c}^{(m)}$  for  $\forall l > m$ . Set  $\mathbf{c}^o = \mathbf{c}^{(m)}$ , from (17), we have  $\mathbf{c}^o \prec \mathbf{c}^*$ . In addition,  $\mathbf{c}^*$  is the smallest dual-optimal and  $\mathbf{c}^o$  is the non-negative fix-point, then we get  $\mathbf{c}^* \prec \mathbf{c}^o$  by Lemma 2.2 and 2.4. Therefore,  $\mathbf{c}^* = \mathbf{c}^o$  and  $\mathbf{d}^* = \mathcal{D}(\mathbf{c}^*)$ . Note that the sequence  $\|\mathbf{c}^{(k)}\|_1$  increases by 1 on each iteration at most. So the Algorithm 1 finds the smallest dual-optimal pair  $\mathbf{c}^*$  and  $\mathbf{d}^*$  by at most  $\|\mathbf{c}^*\|_1 + 1$  iterations.  $\square$

Now, we are able to give the main theorem below.

**Theorem 2.6** *If the  $\Sigma$  matrix of given DAE systems in Problem 4 contains some transversal  $T$ , then the unique smallest dual-optimal pair  $\mathbf{c}^*$  and  $\mathbf{d}^*$  can be found in time  $O(n^3 + \|\mathbf{c}^*\|_1 \cdot n^2)$  via fixed-point iteration algorithm.*

PROOF. It can be easily proved by Lemma 2.4 and 2.5.  $\square$

**Example 2.1** *Consider the application of the Algorithm 1 to nonlinear DAE system  $\mathbf{f} = (f_1, f_2, f_3) = \mathbf{0}$  in three dependent variables  $x_1(t), x_2(t), x_3(t)$  with known forcing functions  $u_i(t) (i = 1, 2, 3)$ :*

$$\begin{cases} f_1 = \ddot{x}_1 + x_3 + u_1(t) \\ f_2 = \dot{x}_2 + x_3 + u_2(t) \\ f_3 = x_1^2 + x_2^2 + u_3(t) \end{cases}.$$

The corresponding signature matrix is

$$\Sigma = \begin{pmatrix} 2^* & 0 \\ & 1 & 0^* \\ 0 & 0^* \end{pmatrix},$$

where we have already marked the HVT with asterisks, and the elements in the blanks of  $\Sigma$  are  $-\infty$ . We give the main process of Algorithm 1 below,

$$\Sigma \Rightarrow \begin{array}{ccccc} & x_1 & x_2 & x_3 & \mathbf{c}'^{(0)} \\ f_1 & \begin{bmatrix} 2^* & & 0 \end{bmatrix} & & & 0 \\ f_2 & \begin{bmatrix} & 1 & 0^* \end{bmatrix} & & & 0 \\ f_3 & \begin{bmatrix} 0 & 0^* \end{bmatrix} & & & 0 \end{array} \Rightarrow \begin{array}{ccccc} & x_1 & x_2 & x_3 & \mathbf{c}'^{(0)} & \mathbf{c}^{(0)} \\ f_1 & \begin{bmatrix} 2^* & & 0 \end{bmatrix} & & & 0 & 0 \\ f_2 & \begin{bmatrix} & 1 & 0^* \end{bmatrix} & & & 0 & 0 \\ f_3 & \begin{bmatrix} 0 & 0^* \end{bmatrix} & & & 0 & 1 \\ \mathbf{d}^{(0)} & 2 & 1 & 0 & & \end{array}$$

where  $\mathbf{c}'^{(i)}$ ,  $\mathbf{c}^{(i)}$ , and  $\mathbf{d}^{(i)}$  mean the  $i$ th iteration for  $\mathbf{c}'$ ,  $\mathbf{c}$  and  $\mathbf{d}$ , respectively. Therefore, obtain the smallest offsets  $\mathbf{c} = (0, 0, 1)$  and  $\mathbf{d} = (2, 1, 0)$  of the DAE systems.

### 3 Block fixed-point iteration method

When dealing with DAE systems of large dimensions, an important manipulation is the block triangularization of the system [8], which allows to decompose the overall system into subsystems which can be solved in sequence. Similarly, considering the index reduction for large-scale systems, it is necessary to compute the block triangular forms (BTF) of the  $\Sigma$  matrix by permuting its rows and columns [13, 4].

In this section, assume the given DAE systems are structurally nonsingular, meaning that the  $\Sigma$  matrix of the systems exists a transversal, then obtain the BTF of the  $\Sigma$  matrix below,

$$M = \begin{pmatrix} M_{11} & M_{1,2} & \cdots & M_{1,\ell} \\ & M_{2,2} & \cdots & M_{2,\ell} \\ & & \ddots & \vdots \\ & & & M_{\ell,\ell} \end{pmatrix}, \quad (19)$$

where the elements in the blanks of  $M$  are  $-\infty$ , and diagonal matrix  $M_{i,i}$  is square and irreducible, for  $i = 1, 2, \dots, \ell$  [15].

The main idea of block fixed-point iteration method for  $\Sigma$  matrix with BTF is to use the fixed-point iteration method with parameter mentioned below to process each diagonal matrix in block upper triangulated signature matrix from top to bottom in sequence. We give the fixed-point iteration method with parameter, firstly.

#### 3.1 Fixed-point iteration method with parameter

The Dual Problem 4 with  $n$  dimension nonnegative parameter vector  $\mathbf{p}$  is defined as follows:

**Definition 3.1** *The Dual Problem with nonnegative parameter  $\mathbf{p}$  is defined*

via:

$$\begin{aligned}
\min_{\mathbf{c}, \mathbf{d}} \quad & z = \sum_j d_j - \sum_i c_i, \\
\text{s.t.} \quad & d_j - c_i \geq \sigma_{ij} \text{ for all } (i, j), \\
& d_j \geq p_j, \text{ for all } j, \\
& c_i \geq 0 \text{ for all } i.
\end{aligned} \tag{20}$$

For any given nonnegative parameter  $\mathbf{p}$ , obtain the fixed-point iteration algorithm with parameter (PFPIA) below just by modified the fixed-point iteration algorithm.

---

**Algorithm 2** Fixed-point iteration algorithm with parameter (PFPIA)

---

**Input:**

$\Sigma$  is signature matrix for DAE systems  
 $\mathbf{p}$  is nonnegative parameter vector

**Output:**

$\mathbf{c}$  and  $\mathbf{d}$

- 1: Set  $\mathbf{c}' \leftarrow \mathbf{0}$
  - 2: Set  $T$  is HVT of  $\Sigma$  by Kuhn-Munkres algorithm
  - 3: Set  $\mathbf{c}' \leftarrow \mathcal{C}_T(\mathbf{p})$
  - 4: Set  $\mathbf{c}' \leftarrow \max\{\mathbf{c}', \mathbf{0}\}$
  - 5: Set  $\mathbf{d} \leftarrow \mathcal{D}(\mathbf{c}')$
  - 6: Set  $\mathbf{c} \leftarrow \mathcal{C}_T(\mathbf{d})$
  - 7: **while**  $\mathbf{c} \neq \mathbf{c}'$  **do**
  - 8:   Set  $\mathbf{c}' \leftarrow \mathbf{c}$
  - 9:   Set  $\mathbf{d} \leftarrow \mathcal{D}(\mathbf{c})$
  - 10:   Set  $\mathbf{c} \leftarrow \mathcal{C}_T(\mathbf{d})$
  - 11: **end while**
  - 12: **return**  $\mathbf{c}, \mathbf{d}$
- 

**Lemma 3.1** *Let  $\mathbf{p}$  is any nonnegative parameter. Assume that the  $\Sigma$  matrix in the Problem 20 contains a transversal  $T$ , then there exists a unique smallest dual-optimal pair  $\mathbf{c}^*$  and  $\mathbf{d}^*$  such that*

$$\mathbf{c}^* = \min\{\mathbf{c} | \mathcal{D}(\mathbf{c}) \succ \mathbf{p}, \mathbf{c} \in VC\}.$$

*Moreover, if the used transversal  $T$  is HVT, the Algorithm 2 finds the unique smallest dual-optimal pair  $\mathbf{c}^*$  and  $\mathbf{d}^*$  by at most  $\|\mathbf{c}^*\|_1 - \|\max\{\mathcal{C}_T(\mathbf{p}), \mathbf{0}\}\|_1 + 1$  iterations.*

PROOF. Just modify the proof of Lemma 2.4 and 2.5 properly.  $\square$

**Remark 3.2** If  $p_j \leq \max_i \sigma_{ij}$ , for  $j = 1, 2, \dots, n$ , and  $d_j \geq \max_i \sigma_{ij}$  derived from Problem (20), for  $j = 1, 2, \dots, n$ . So obtain  $d_j \geq p_j$ , for  $j = 1, 2, \dots, n$ , that is, the constraint condition  $d_j \geq p_j$  in Problem (20) can be deleted. Therefore, the Problem (20) turns into Problem (4).

**Example 3.1** Consider the application of the Algorithm 2 to nonlinear DAE systems  $\mathbf{f} = (f_4, f_5, f_6) = \mathbf{0}$  in three dependent variables  $x_4(t), x_5(t), x_6(t)$  with known forcing functions  $u_i(t) (i = 4, 5, 6)$ :

$$\begin{cases} f_4 = \ddot{x}_4 + x_6 + u_4(t) \\ f_5 = \dot{x}_5 + x_6 + u_5(t) , \\ f_6 = x_4^2 + x_5^2 \end{cases}$$

and the any given parameter is  $\mathbf{p} = (0, 0, 2)$ .

We give below the main process of Algorithm 2,

$$\begin{aligned} \Sigma \Rightarrow \begin{array}{c} \mathbf{p} \quad 0 \quad 0 \quad 2 \\ \quad x_4 \quad x_5 \quad x_6 \quad \mathbf{c}'^{(0)} \\ f_4 \begin{bmatrix} 2^* & & 0 \end{bmatrix} & 0 \\ f_5 \begin{bmatrix} & 1 & 0^* \end{bmatrix} & 2 \\ f_6 \begin{bmatrix} 0 & 0^* & \end{bmatrix} & 0 \end{array} \quad \Rightarrow \quad \begin{array}{c} \mathbf{p} \quad 0 \quad 0 \quad 2 \\ \quad x_4 \quad x_5 \quad x_6 \quad \mathbf{c}'^{(0)} \quad \mathbf{c}^{(0)} \\ f_4 \begin{bmatrix} 2^* & & 0 \end{bmatrix} & 0 & 0 \\ f_5 \begin{bmatrix} & 1 & 0^* \end{bmatrix} & 2 & 2 \\ f_6 \begin{bmatrix} 0 & 0^* & \end{bmatrix} & 0 & 3 \\ \mathbf{d}^{(0)} & 2 & 3 & 2 \end{array} \\ \\ \Rightarrow \begin{array}{c} \mathbf{p} \quad 0 \quad 0 \quad 2 \\ \quad x_4 \quad x_5 \quad x_6 \quad \mathbf{c}'^{(0)} \quad \mathbf{c}^{(0)} \quad \mathbf{c}^{(1)} \\ f_4 \begin{bmatrix} 2^* & & 0 \end{bmatrix} & 0 & 0 & 1 \\ f_5 \begin{bmatrix} & 1 & 0^* \end{bmatrix} & 2 & 2 & 2 \\ f_6 \begin{bmatrix} 0 & 0^* & \end{bmatrix} & 0 & 3 & 3 \\ \mathbf{d}^{(0)} & 2 & 3 & 2 \\ \mathbf{d}^{(1)} & 3 & 3 & 2 \end{array} \quad \Rightarrow \quad \begin{array}{c} \mathbf{p} \quad 0 \quad 0 \quad 2 \\ \quad x_4 \quad x_5 \quad x_6 \quad \mathbf{c}'^{(0)} \quad \mathbf{c}^{(0)} \quad \mathbf{c}^{(1)} \quad \mathbf{c}^{(2)} \\ f_4 \begin{bmatrix} 2^* & & 0 \end{bmatrix} & 0 & 0 & 1 & 1 \\ f_5 \begin{bmatrix} & 1 & 0^* \end{bmatrix} & 2 & 2 & 2 & 2 \\ f_6 \begin{bmatrix} 0 & 0^* & \end{bmatrix} & 0 & 3 & 3 & 3 \\ \mathbf{d}^{(0)} & 2 & 3 & 2 \\ \mathbf{d}^{(1)} & 3 & 3 & 2 \\ \mathbf{d}^{(2)} & 3 & 3 & 2 \end{array} , \end{aligned}$$

where  $\mathbf{c}'^{(i)}$ ,  $\mathbf{c}^{(i)}$  and  $\mathbf{d}^{(i)}$  mean the  $i$ th iteration for  $\mathbf{c}'$ ,  $\mathbf{c}$  and  $\mathbf{d}$ , respectively. Then obtain the smallest offsets  $\mathbf{c} = (1, 2, 3)$  and  $\mathbf{d} = (3, 3, 2)$  for the DAEs.

### 3.2 Block fixed-point iteration method

The given DAE systems are structurally nonsingular, obtain the  $\Sigma$  matrix  $M$  of Problem 4 with block triangular form (19), and  $\sum_{i=1}^{\ell} n_i = n$ , where  $n_i$  is the order of  $M_{ii}$ ,  $i = 1, 2, \dots, \ell$ . In order to find the unique smallest dual-optimal, we give some necessary symbols and definitions as follows.

Let the parameter vector is  $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_\ell)$  with  $\ell$  sections, the dual-optimal are  $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_\ell)$  and  $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_\ell)$ , where the dimension of  $\mathbf{p}_i$ ,  $\mathbf{c}_i$  and  $\mathbf{d}_i$  are  $n_i$  for  $i = 1, 2, \dots, \ell$ . For any  $n \times r$  order matrix  $B$  and  $B'$ ,  $n$  order vector  $\mathbf{q}$ ,  $\bar{\mathbf{q}}$  and  $\hat{\mathbf{q}}$ ,  $r$  order vector  $\mathbf{w}$ , the mapping  $B' = \mathbf{RowAdd}(B, \mathbf{q})$  is defined as  $B'_{i,j} = B_{i,j} + q_i$ , for  $i = 1, 2, \dots, n, j = 1, 2, \dots, r$ ; the mapping  $\mathbf{w} = \mathbf{ColMax}(B)$  is defined via  $w_j = \max_{i \in \{1, 2, \dots, n\}} B_{i,j}$ , for  $j = 1, 2, \dots, r$ ; the mapping  $\mathbf{q} = \mathbf{eMax}(\bar{\mathbf{q}}, \hat{\mathbf{q}})$  is defined as  $q_i = \max(\bar{q}_i, \hat{q}_i)$ ,  $i = 1, 2, \dots, n$ . Then we give block fixed-point iteration algorithm below.

---

#### Algorithm 3 Block fixed-point iteration algorithm

---

**Input:**

$M$  is  $\Sigma$  matrix of given DAE systems with BTF (19)

**Output:**

$\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_\ell)$  and  $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_\ell)$

- 1: Set  $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_\ell)$ ,  $\mathbf{p}_j = \mathbf{0}, j = 1, 2, \dots, \ell$ .
- 2: Set  $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_\ell)$ ,  $\mathbf{c}_j = \mathbf{0}, j = 1, 2, \dots, \ell$ .
- 3: Set  $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_\ell)$ ,  $\mathbf{d}_j = \mathbf{0}, j = 1, 2, \dots, \ell$ .
- 4: Get  $(\mathbf{c}_1, \mathbf{d}_1) = \text{PFPIA}(M_{11}, \mathbf{p}_1)$ .
- 5: **for**  $i$  from 2 to  $\ell$  **do**
- 6:   Update:  $[M_{i-1,i}, \dots, M_{i-1,\ell}] \leftarrow \mathbf{RowAdd}([M_{i-1,i}, \dots, M_{i-1,\ell}], \mathbf{c}_{i-1})$ .
- 7:   Update:  $\mathbf{p}_i \leftarrow \mathbf{ColMax}\left(\begin{pmatrix} M_{1,i} \\ \vdots \\ M_{i-1,i} \end{pmatrix}\right)$ .
- 8:   Update:  $\mathbf{p}_i \leftarrow \mathbf{eMax}(\mathbf{p}_i, \mathbf{0})$ .
- 9:   Get  $(\mathbf{c}_i, \mathbf{d}_i) = \text{PFPIA}(M_{ii}, \mathbf{p}_i)$ .
- 10: **end for**
- 11: **return**  $\mathbf{c}, \mathbf{d}$

---

In order to obtain a complete theoretical analysis of block fixed-point iteration method, we give some necessary lemmas, firstly.

**Lemma 3.3** ([15]). *For given  $\Sigma$  matrix  $M$  with BTF (19), if  $T_i$  is HVT of  $M_{ii}$ , then  $T = \bigcup_{i=1}^{\ell} T_i$  is HVT of  $M$ .*

**Lemma 3.4** *Assume that the Problem 4 of DAE systems is structurally non-singular, then fixed-point iteration algorithm gives the same smallest dual-optimal pair with block fixed-point iteration algorithm.*

PROOF. Without loss of generality, assume that the  $\Sigma$  matrix  $M$  of Problem 4 is block triangular forms (19). Set  $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_\ell)$  with  $\ell$  sections is the parameter vector;  $\mathbf{c}^o = (\mathbf{c}_1^o, \mathbf{c}_2^o, \dots, \mathbf{c}_\ell^o)$  and  $\mathbf{d}^o = (\mathbf{d}_1^o, \mathbf{d}_2^o, \dots, \mathbf{d}_\ell^o)$  are the smallest dual-optimal found by block fixed-point iteration algorithm;  $\mathbf{c}^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, \dots, \mathbf{c}_\ell^*)$  and  $\mathbf{d}^* = (\mathbf{d}_1^*, \mathbf{d}_2^*, \dots, \mathbf{d}_\ell^*)$  are the smallest dual-optimal found by fixed-point iteration algorithm.

For  $\ell$  is integer, we prove the lemma by mathematical induction. Considering about  $\ell = 1$ , it is easy to know  $\mathbf{p}_1 = \mathbf{0}$ , so then  $\mathbf{c}^o = \mathbf{c}^*$  and  $\mathbf{d}^o = \mathbf{d}^*$ , that is, the lemma is true. Assume the lemma is true when  $\ell = N - 1$ , that is,

$$\mathbf{c}_k^o = \mathbf{c}_k^*, \quad \mathbf{d}_k^o = \mathbf{d}_k^*, \quad (21)$$

for  $k = 1, 2, \dots, N - 1$ . We now consider about  $\ell = N$ . From (21), obtain

$$\begin{aligned} \mathbf{p}_N &= \mathbf{eMax}(\mathbf{ColMax}(\mathbf{RowAdd}(\begin{pmatrix} M_{1,N} \\ \vdots \\ M_{N-1,N} \end{pmatrix}, \begin{pmatrix} \mathbf{c}_1^o \\ \vdots \\ \mathbf{c}_{N-1}^o \end{pmatrix})), \mathbf{0}) \\ &= \mathbf{eMax}(\mathbf{ColMax}(\mathbf{RowAdd}(\begin{pmatrix} M_{1,N} \\ \vdots \\ M_{N-1,N} \end{pmatrix}, \begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{N-1}^* \end{pmatrix})), \mathbf{0}). \end{aligned} \quad (22)$$

It is easily verified that  $\mathbf{c}_N^*$  and  $\mathbf{d}_N^*$  are the dual-optimal of Problem 20 with parameter  $\mathbf{p}_N$  by (22). From block fixed-point iteration algorithm, note that

$$\mathbf{c}_N^o = \min\{\mathbf{c} | \mathcal{D}(\mathbf{c}) \succ \mathbf{p}_N, \mathbf{c} \in VC\}. \quad (23)$$

From Lemma 3.1 and (23), we obtain

$$\mathbf{c}_N^o \prec \mathbf{c}_N^*, \quad \mathbf{d}_N^o = \mathcal{D}(\mathbf{c}_N^o) \prec \mathbf{d}_N^* = \mathcal{D}(\mathbf{c}_N^*). \quad (24)$$

On the other hand, construct  $\mathbf{c}^{*o} = (\mathbf{c}_1^*, \dots, \mathbf{c}_{N-1}^*, \mathbf{c}_N^o)$  and  $\mathbf{d}^{*o} = (\mathbf{d}_1^*, \dots, \mathbf{d}_{N-1}^*, \mathbf{d}_N^o)$ . Note that  $\mathbf{c}_N^o$  and  $\mathbf{d}_N^o$  are the smallest dual-optimal of Problem 20 with parameter  $\mathbf{p}_N$ . Moreover, from (21) and (22),  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$  are the dual feasible solution of Problem 4. By Lemma 2.1(iii), obtain

$$\mathbf{d}_{k j_k}^* - \mathbf{c}_{k i_k}^* = \sigma_{i_k, j_k}, \quad (25)$$

for each  $(i_k, j_k) \in T_k$ ,  $k = 1, 2, \dots, N-1$ , and

$$\mathbf{d}_{N j_N}^o - \mathbf{c}_{N i_N}^o = \sigma_{i_N, j_N} \quad (26)$$

for each  $(i_N, j_N) \in T_N$ . From (25,26) and Lemma 3.3, we have

$$\mathbf{d}_j^{*o} - \mathbf{c}_i^{*o} = \sigma_{i,j} \quad (27)$$

for each  $(i, j) \in T = \bigcup_{i=1}^N T_i$ . That is,  $\mathbf{c}^{*o}$  and  $\mathbf{d}^{*o}$  are dual-optimal of Problem 4 with  $\Sigma = M$  by Lemma 2.1(iii). Note that  $\mathbf{c}^*$  and  $\mathbf{d}^*$  are the smallest dual-optimal of Problem 4. By Lemma 2.4, obtain

$$\mathbf{c}^* \prec \mathbf{c}^{*o}, \mathbf{d}^* \prec \mathbf{d}^{*o}. \quad (28)$$

Moreover, by (28), we get

$$\mathbf{c}_N^* \prec \mathbf{c}_N^o, \mathbf{d}_N^* \prec \mathbf{d}_N^o. \quad (29)$$

Combining (24) with (29), obtain

$$\mathbf{c}_N^o = \mathbf{c}_N^*, \mathbf{d}_N^o = \mathbf{d}_N^*. \quad (30)$$

So we have  $\mathbf{c}^* = \mathbf{c}^o$  and  $\mathbf{d}^* = \mathbf{d}^o$ . It is shown that the lemma is true.  $\square$

Now, we are able to obtain the main theorem as follows.

**Theorem 3.5** *If the  $\Sigma$  matrix for given DAE systems with BTF(19) is structurally nonsingular, then the unique smallest dual-optimal pair  $\mathbf{c}^* = (\mathbf{c}_1^*, \mathbf{c}_2^*, \dots, \mathbf{c}_\ell^*)$  and  $\mathbf{d}^*$  of the DAE can be found in time  $O(\sum_{i=1}^\ell n_i^3 + \|\mathbf{c}_i^*\|_1 \cdot n_i^2)$  by block fixed-point iteration algorithm. Furthermore, if  $n_i = r$  for each  $i$ , i.e.,  $n = \ell \cdot r$ , then the time is  $O(\ell \cdot r^3 + \|\mathbf{c}^*\|_1 \cdot r^2)$ .*

PROOF. We can easily prove the theorem by Theorem 2.6 and Lemma 3.1 and 3.4.  $\square$

**Example 3.2** Consider the application of the Algorithm 3 to nonlinear DAE system  $\mathbf{f} = (f_1, f_2, \dots, f_6) = \mathbf{0}$  in six dependent variables  $x_1(t), x_2(t), \dots, x_6(t)$  with known forcing functions  $u_i(t) (i = 1, 2, \dots, 6)$ :

$$\begin{cases} f_1 = \ddot{x}_1 + x_3 + u_1(t) \\ f_2 = \dot{x}_2 + x_3 + u_2(t) \\ f_3 = x_1^2 + x_2^2 + \dot{x}_6 + u_3(t) \\ f_4 = \ddot{x}_4 + x_6 + u_4(t) \\ f_5 = \dot{x}_5 + x_6 + u_5(t) \\ f_6 = x_4^2 + x_5^2 + u_6(t) \end{cases}.$$

The corresponding signature matrix is

$$\Sigma = \begin{array}{c} \mathbf{p} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 2 \\ \quad \quad x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6 \quad \tilde{\mathbf{c}}^* \quad \mathbf{c}^* \\ \begin{array}{c} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ \tilde{\mathbf{d}}^* \\ \mathbf{d}^* \end{array} \left( \begin{array}{ccc|cc} 2^* & & 0 & & & \\ & 1 & 0^* & & & \\ 0 & 0^* & & & & 1 \\ \hline & & & 2^* & & 0 \\ & & & & 1 & 0^* \\ & & & 0 & 0^* & \\ \hline 2 & 1 & 0 & 2 & 1 & 0 \\ 2 & 1 & 0 & 3 & 3 & 2 \end{array} \right) \begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \end{array},$$

where we have already marked the *HVT* with asterisks;  $\mathbf{p} = (0, 0, 0, 0, 0, 2)$  is the corresponding parameter vector by block fixed-point iteration algorithm;  $\tilde{\mathbf{c}}_i^* = (0, 0, 1)$  and  $\tilde{\mathbf{d}}_i^* = (2, 1, 0)$  are the local smallest offsets for each diagonal signature matrix  $\Sigma_{ii}$ ,  $i = 1, 2$  via fixed-point iteration algorithm;  $\mathbf{c}^* = (0, 0, 1, 1, 2, 3)$  and  $\mathbf{d}^* = (2, 1, 0, 3, 3, 2)$  are the global smallest offsets for  $\Sigma$  matrix directly using fixed-point iteration algorithm.

In the following, the main process of block fixed-point iteration algorithm is shown. The signature matrix  $\Sigma$  above contains two blocks. For  $i = 1$ ,  $\mathbf{p}_1 = (0, 0, 0)$  is the parameter for the first diagonal block, get  $\mathbf{c}_1^* = \tilde{\mathbf{c}}_1^* = (0, 0, 1)$  and  $\mathbf{d}_1^* = \tilde{\mathbf{d}}_1^* = (2, 1, 0)$  from Example 2.1. For  $i = 2$ , we obtain  $\mathbf{p}_2 = (0, 0, 2)$  which is the parameter for the second diagonal block, and obtain  $\mathbf{c}_2^* = (1, 2, 3)$ ,  $\mathbf{d}_2^* = (3, 3, 2)$  from Example 3.1. So  $\mathbf{c}^* = (\mathbf{c}_1^*, \mathbf{c}_2^*) = (0, 0, 1, 1, 2, 3)$  and  $\mathbf{d}^* = (\mathbf{d}_1^*, \mathbf{d}_2^*) = (2, 1, 0, 3, 3, 2)$  are the smallest offsets for the DAE system.



## 4 Conclusions

In this paper, we reinforce the theoretical foundation for Pryce's structural index reduction method of DAE systems, finely prove the existence and uniqueness of the smallest offsets, and then show the polynomial complexity for finding optimal index reduction for given DAEs.

To solve large scale DAE systems with block structure, we describe a block fixed-point iteration method which can be applied to a sequence of sub-systems rather than the whole system. Accordingly, the time complexity of our method decreases proportionally with the number of the diagonal blocks in the signature matrix.

As pointed in the Campbell-Griepentrog Robot Arm [3] and the special DAE with parameter [6], Pryce's structural analysis method fails to find a DAE's true structure because of producing an identically singular Jacobian. What's more, for a class of simple DAE systems with special  $n \times n$  signature matrix  $\Sigma$  [17], the actual number of iterations of fixed-point iteration algorithm (i.e.,  $O(n)$ ) is significantly less than  $\|\mathbf{c}^*\|_1$ . We believe these situations have appeared rarely in the practical applications. Compared with other structural index reduction methods, our method can address a fairly wide class of large-scale DAE systems precisely and efficiently. And the actual performance of block fixed-point iteration algorithm will be discussed in future work.

## Acknowledgements

This work is partially supported by China 973 Project (Grant No. NKBRPC-2011CB302402), National Natural Science Foundation of China (Grant Nos. 11471307, 61402537, 11171053, 91118001), the West Light Foundation of the Chinese Academy of Sciences from China, the project of Chongqing Science and Technology Commission from China (Grant No. cstc2013jjys40001).

## References

- [1] K.E. Brenan, S.L. Campbell and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential Algebraic Equations*, SIAM Publications, Philadelphia, PA, 2nd edition, 1996.

- [2] S.L. Campbell and C.W. Gear, *The index of general nonlinear DAEs*, Numerische Mathematik, 72 (1995), pp. 173–196.
- [3] S.L. Campbell and E. Griepentrog, *Solvability of general differential algebraic equations*, SIAM J. Sci. Comput., 16(2) (1995), pp. 257–270.
- [4] I.S. Duff and B. Uçar, *On the block triangular form of symmetric matrices*, SIAM Review, 53(3) (2010), pp. 455–470.
- [5] C.W. Gear and L.R. Petzold, *ODE methods for the solution of differential/algebraic systems*, SIAM J. Numer. Anal., 21(4) (1983), pp. 716–728.
- [6] R. Lamour and R. März, *Detecting structures in differential algebraic equations: Computational aspects*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 4055–4066.
- [7] R. Lamour, C. Tischendorf and R. März, *Differential-Algebraic Equations: A Projector Based Analysis*, Springer-Verlag Berlin Heidelberg, 2013.
- [8] C. Maffezzoni, R. Girelli and P. Lluka, *Generating efficient computational procedures from declarative models*, Simulation Practice and Theory, 4 (1996), pp. 303–317.
- [9] Modelica Association, *Modelica-A unified object-oriented language for systems modeling: language specification*, 2012. Available at <https://www.modelica.org/documents/ModelicaSpec33.pdf>,
- [10] N.S. Nedialkov, J.D. Pryce and G.N. Tan, *DAESA-A matlab tool for structural analysis of DAEs: Software*, preprint (2014), to appear in ACM Transactions on Mathematical Software. Available at [http://www.cas.mcmaster.ca/~nedialk/PAPERS/DAEs/daesa\\_software/daesaSoftware.pdf](http://www.cas.mcmaster.ca/~nedialk/PAPERS/DAEs/daesa_software/daesaSoftware.pdf).
- [11] C.C. Pantelides, *The consistent initialization of differential-algebraic systems*, SIAM J. Sci. Stat. Comput., 9(2) (1998), pp. 213–231.
- [12] L.R. Petzold, *Differential/algebraic equations are not ODEs*, SIAM J. Sci. Stat. Comp., 3 (1982), pp. 367–384.
- [13] A. Pothén and C. Fan, *Computing the block triangular form of a sparse matrix*, ACM Transactions on Math. Softw., 16(4) (1990), pp. 303–324.
- [14] J.D. Pryce, *A simple structural analysis method for DAEs*, BIT, 41(2) (2001), pp. 364–394.
- [15] J.D. Pryce, N.S. Nedialkov and G.N. Tan, *DAESA-A matlab tool for structural analysis of DAEs: Theory*, preprint (2014), to appear in ACM Transactions on Mathematical Software. Available at <http://www.cas.mcmaster.ca/cas/0reports/CAS-12-01-NN.pdf>.
- [16] X.L. Qin, W.Y. Wu, Y. Feng and G. Reid, *Structural analysis of high index DAE for process simulation*, Int. J. Model. Simul. Sci. Comput., 4(4) (2013). doi:10.1142/S1793962313420087.

- [17] G. Reissig, W.S. Martinson and P.I. Barton, *Differential-algebraic equations of index 1 may have an arbitrarily high structural index*, SIAM J. Sci. Comput., 21(6) (2000), pp. 1987–1990.
- [18] W.C. Rheinboldt, *Differential-algebraic systems as differential equations on manifolds*, Math. Comp., 43 (1984), pp. 473–482.
- [19] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer-Verlag Berlin Heidelberg, 2004.
- [20] W.Y. Wu, G. Reid and S. Ilie, *Implicit Riquier Bases for PDAE and their Semi-Discretizations*, J. Symb. Comput., 44 (2009), pp. 923–941.